Activation Functions in Machine Learning

Activation Functions

Activation functions are used in Neural Networks to facilitate the models with non-linearity relationship of the input and the output. If activation functions are not used in Artificial Neural Networks, then no matter how many hidden layers we choose, the resulting model would simply be as good as a set of linear regression models without any nonlinearity provision. Five commonly used activation functions: Linear, Sigmoid, Hyperbolic Tangent (Tanh), ReLU, and Softmax are explained below.

Linear Activation

Description: The Linear activation function is a simple identity function. It is as good as no activation function. It is mainly used in the output layer of regression models (and sometimes used with from_logits=True in TensorFlow to make programmatically efficient computations in a multi-class classification problem).

Formula:

$$f(x) = x$$

Output range: $(-\infty, \infty)$

$$f'(x) = 1$$

Sigmoid Activation

Description: The Sigmoid function transforms the input into a range between 0 and 1, which is used especially for binary classification tasks.

Formula:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Output range: $(0, 1)$

Derivative:

$$f'(x) = f(x) \cdot (1 - f(x))$$

Hyperbolic Tangent (Tanh) Activation

Description: The Tanh function outputs values between -1 and 1, which helps in centering the data that may lead to faster convergence.

Formula:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Output range: $(-1, 1)$

Derivative:

$$f'(x) = 1 - f(x)^2$$

Rectified Linear Unit (ReLU) Activation

Description: ReLU is widely used in deep neural networks due to its simplicity and effectiveness. Due to its discontinuity, several layers with ReLU activation function could effectively facilitate nonlinear relationship between inputs and outputs.

Formula:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Output range: $[0, \infty)$
Derivative:

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Softmax Activation

Description: Softmax is used in the output layer for multi-class classification. It is applied on a vector (instead of a scalar) and the result would be a vector containing the probabilities that sum up to 1. Whichever has the maximum probability is inferred to be the predicted class.
Formula:

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Output range: $(0, 1)$, with $\sum_i f(x_i) = 1$
Derivative:

$$\frac{\partial f(x_i)}{\partial x_j} = \begin{cases} f(x_i) \cdot (1 - f(x_j)) & \text{if } i = j \\ -\left(f(x_i)\right)^2 & \text{if } i \neq j \end{cases}$$

The derivative of the whole softmax-activated vector with respect to each of the variables would be a matrix.